Computer Science
Technical Report

Colorado State University

Interleaving 3D Model Feature Prediction and Matching to Support Multi-Sensor Object Recognition *

Mark R. Stevens and J. Ross Beveridge

December 16, 1995

Technical Report CS-96-107

Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873

Phone: (970) 491-5792   Fax: (970) 491-2466
WWW: http://www.cs.colostate.edu

*This work was sponsored by the Advanced Research Projects Agency (ARPA) under grant DAAH04-93-G-422, monitored by the U.S. Army Research Office.

Interleaving 3D Model Feature Prediction and Matching to Support Multi-Sensor Object Recognition *†

Mark R. Stevens
Colorado State University
stevensm@cs.colostate.edu

J. Ross Beveridge
Colorado State University
ross@cs.colostate.edu

Abstract

Recognizing 3D modeled objects through alignment of object and sensor features requires a means of predicting matchable features. This paper presents a system which performs on-line feature prediction for CAD modeled objects and tightly couples prediction with matching. For the ATR domain, detailed CAD models of objects are available in this application, as is both range and optical imagery. Matching begins with an initial hypothesis which is refined through an iterative generate-and-test procedure. Matching interleaves feature prediction and adjustment of model-to-sensor geometry until a locally optimal match is obtained. In addition, sensor-to-sensor geometry is also adjusted, allowing the algorithm to correct minor mis-registrations between range and optical imagery. While the resulting match is locally optimal in terms of the complete space of possible matches, it globally preserves the 3D constraints implied by sensor and object geometry. Results on real data are presented which demonstrate the algorithm correcting for up to 30° errors in initial orientation and $25m$ errors in initial translation.

1 Introduction

Detailed CAD models offer rich geometric constraints for object recognition. However, the object model itself is seldom in a form suitable for direct matching to image features. Significant steps must be taken to map from the stored model representation to features likely to be detectable. This paper presents both a feature prediction algorithm and a local search matching algorithm which utilizes this prediction capability to refine features during matching in a generate-and-test fashion.

*This work was sponsored by the Advanced Research Projects Agency (ARPA) under grant DAAH04-93-G-422, monitored by the U.S. Army Research Office.

†Appears also in the Proceedings of the 1996 ARPA Image Understanding Workshop.

Common approaches to model feature prediction have focused upon developing off-line data structures which capture feature visibility information associated with geometry alone [PD87, Pla88, Ike87]. This process usually begins with the division of all possible viewpoints into sets of constant model topology [KvD76, KvD79]. From these regions, silhouette and other model features can be determined and stored [SD92, KD87] for later retrieval during matching. Finally, some promising recent work has used statistical modeling to predict feature visibility based upon both geometry and lighting [PHK91, WI93].

In contrast to much of this work, we are promoting an on-line prediction capability which performs the mapping from stored model to predicted features dynamically as part of the recognition process. A key to making this approach feasible is the development of algorithms which run many, if not all, computations in parallel on standard graphics acceleration hardware. This on-line capability permits us to develop a tight coupling between feature prediction and matching: modifying the features expected to be visible as matching progresses.

The algorithms presented here are being developed to perform final verification within a larger Automatic Target Recognition (ATR) system [BHP95]. Thus, upstream color-detection [BDHR94] and range boundary matching algorithms [Bev92] provide hypotheses indicating a specific target is at roughly the following position and orientation relative to the sensor platform. Consequently, the primary aim of the matching algorithm presented here is to reliably refine the pose estimate and match between object model and sensor features.

In this domain, both range and optical imagery is available. The addition of range data is extremely helpful, since ATR problems are typically more difficult than other commonly studied object recognition problems. Often complex objects are viewed at great distances, in scenes where backgrounds may contain significant amounts of clutter. Vehicles usually blend in well with the surroundings and may be partially obscured.

While having both range and optical imagery is helpful, the integration of these two heterogeneous sensors introduces an image registration problem. In an ideal world, the registration mapping between sensors could be uniquely determined through off-line calibration. Unfortunately, such estimates are usually only accurate to within several pixels. Thus, in addition to refining the estimated 3D pose of the object relative to the sensor suite, our algorithm also refines the pixel-to-pixel registration estimate between the range and optical sensors. We use the term *coregistration* to describe this combined process of simultaneously adjusting object pose and sensor registration estimates.

The remaining portion of the paper is divided into three main parts: a detailed discussion of the feature prediction algorithm, how feature prediction is used by the local search, and results of the approach on two images.

2 3D Model Feature Prediction

To achieve *coregistration* of an object model to optical and range imagery, model features suitable for matching must first be extracted from the CAD model. The role of prediction is to select which features to extract. Here, 3D line segments are extracted for matching to optical imagery. For range imagery, the choice is elementary: surfaces visible from the estimated viewpoint are sampled.

For optical imagery, selection must not only take into account the issue of physical visibility, but of expected lighting as well. Viewing angle alone is sufficient to determine which model features generate the object silhouette. Since it is assumed that silhouette features are relatively likely to stand out against the background, they are extracted. However, using only silhouettes leads to ambiguity in the matching, and therefore features representing internal detail are extracted as well. Which of the many possible internal features to select is based upon a simple lighting model.

To test our algorithms on real data, we have range, color and IR imagery which we and Martin Marietta collected at Fort Carson, Colorado in November 1993 [BPY94]. The data contains many different image triplets, out of which two pairs of range and color images are used for demonstration here. The first image set, Figure 2, is a simple proof-of-concept image in which the vehicle is roughly $50m$ away in a fairly open area. The second image set, Figure 3, is considerably more difficulty in that the vehicle is approximately $100m$ away on a hillside.

Highly detailed models of the vehicles in our Fort Carson dataset exist in the CAD model format known as BRL/CAD [U. 91]. Algorithms to reduce the model complexity to a level more closely related to the sensor granularity have already been developed [SBG95, Ste95]. From these simpler models, features to be used in the matching process are then obtained. Currently, we have models for an M113 APC and an M60. This paper deals only with the M113, but work has been done matching the M60 [BSS96].

2.1 Predicting 3D Line Segments

The silhouette of an object is a valuable recognition cue when dealing with two-dimensional optical imagery [Mar77, Koe84]. Many systems have been developed to recognize 3D objects based on their projected 2D silhouettes [WW80, LT90, WMA84], and while work using the 3D edges directly is rare [CA87] it is usually concerned with linking 2D image features to 3D model features. Our method approaches the problem from the other direction: we are tying the 3D model edges to the 2D image data.

Hoogs has noted that there exist several forms of contextual information which can be exploited when tackling computer vision problems: geometric, temporal, functional, radiometric, and image context [HH94]. In particular, he has developed a statistical framework for estimating the probability that a given edge will be distinctive enough to be found in the sensor imagery. Our early experiments using only silhouette lines in our domain suggest there is too much ambiguity for the silhouette to adequately constrain the match. Since others have observed improved performance when internal edge structure is added [CSR93, CS94], our feature prediction utilizes simple radiometric and temporal context information in order to predict the internal structure likely to be visible in the optical imagery.

2.1.1 Silhouette Lines

To determine which parts of the CAD model produce the silhouette, a unique color is first assigned to each existing face. This color acts as an index into a hash table of 3D faces. The model is then rendered from the hypothesized viewing orientation. Rendering is performed on a hardware Z-buffer, and hence can be done very quickly. Running on a Sparc 10 with a ZX accelerator, this process takes roughly 0.3 seconds for a model containing 250 faces. The colors of the resulting pixels indicate which faces are visible. Pixels adjacent to the background color, which is also unique, contribute to the model silhouette. Thus, if the background color appears in a pixel's eight-connected neighborhood, the associated face lies on the silhouette.

Subsequent search determines which specific face boundaries (edges) generate the silhouette. An edge is a possible silhouette edge only if one of the two bounding faces is visible [SD92]. This step may leave some edges which are actually internal as hypothesized silhouette edges, and it also does not deal with self-occlusion. A clipping algorithm is then used to discover and discard those edges and portions of edges which are not part of the silhouette. The clipping process projects the 3D model edge endpoints onto the image plane. A line following algorithm then traverses the segment to find the parametric end-points which correspond to the beginning and ending portion of the silhouette edge. Because an orthographic projection is used to render the model, parametric end-point values may be applied directly to the corresponding model edges to produce the resulting 3D silhouette edges.

2.1.2 Internal Lines

To determine if an edge is likely to cause a significant change in illumination in an image, an estimate of the location of the major light source, the sun in our images, must be made available to the feature prediction algorithm. The sun is modelled as an area light source, and the vector to the sun is calculated using a long/lat estimate, time of day, date, and compass orientation [PP76]. All of this information is available for our current data set. Once the vector is determined, it provides the direction to the sun for the entire scene, and can be used to predict the internal model edges.

The internal edge prediction is run after the silhouette extraction phase, and therefore all visible faces are known. The sun vector is then rotated into the proper compass orientation, and the dot product of the vector with the normal of each face is determined. Each edge of the visible faces is then examined independently, and marked as being a possibly significant internal edge. Of this list of possible internal edges, each face which shares the edge is examined and if the predetermined dot products of the two faces with the sun vector are of the same sign, the edge is removed from this list. This simple test determines edges for which light will be cast onto only one of its visible faces.

The final pass of the algorithm uses a clipping algorithm similar to that used in obtaining silhouettes: the 3D edge endpoints are projected onto the image and the parametric endpoint values are determined. The only difference is that our process does not require the edge to lie on the silhouette, that one of its faces needs to be visible. After both silhouette and internal edges are determined for a given pose hypothesis, shorter lines are discarded using a user specified minimum distance threshold. Figures 2b and 2e show the silhouette and internal edges used in the matching process for the pose hypothesis given in Figures 2a and 2d.

2.2 Predicting Sampled Surfaces

A 3D sampled surface is generated in a manner which, in simple terms, simulates the operation of the actual range sensor. The CAD model is transformed into the range sensor's coordinate system using the current estimate of the target position and orientation. Based on the characteristics of the range device, rays are cast into the scene and intersected with the 3D faces of the CAD model. The results of the rendering step used to extract the silhouette are used here to limit ray intersections to only those faces known to be visible. The closest face intersection is stored as the depth of the current position. By design, noise factors are neglected when generating model features: the intention is to generate a high quality model. Noise is dealt with later when matching the model features to the sensor features. Figures 2c and 2f show the sampled surface information used in the matching process for the pose hypothesis given in Figures 2a and 2d.

3 Local Search to Achieve Coregistration

To make use of model features believed to be visible, two items must first be developed: an error function relating model features to the data, and a search mechanism for finding a pose estimate which minimizes that error. In addition to defining the most desirable match, the error function also directs the search process by suggesting local improvements to a current 'best' match.

The match error function which defines the quality of a match $\mathcal{M}$ may be written as:

$$E_{\mathcal{M}}(c, \mathcal{F}) > 0 \quad \forall c \in C,\ \mathcal{F} \in \Re^8 \tag{1}$$

The first argument to this function, c, represents a particular correspondence mapping between model and sensor features. The second argument, $\mathcal{F}$, represents the coregistration of the sensors relative to the model. In the most general case, the correspondence space C is the super-set of all possible pairs of sensor and model features.

Observe that C denotes the set of correspondences between the model features and features derived from both range and optical imagery. This, c, represents the pairing between corresponding sampled surface features from the target model and sensed range points in the LADAR imagery. The mapping c also indicates the correspondence between line segments and the optical imagery.

The coregistration, $\mathcal{F}$, represents the geometric relationship between the sensors and the model. In the development below, this is an eight place vector: six values encode the pose of the target relative to the optical sensor (3 rotation and 3 translation), and two values encode the planar translation of the optical image plane relative to the range sensor's image plane. A detailed justification of this particular parameterization appears in [J. 96].

Two different matching strategies emerge depending on whether the search is conducted in the space of correspondences, C, or the space of coregistration parameters: $\mathcal{F} \in \Re^8$. Searching in correspondence space for a c^* which minimizes equation 1, an optimal coregistration estimate for a given correspondence, c, may be determined using a non-linear least-squares procedure [SB94]. Searching in coregistration space, given a coregistration estimate, $\mathcal{F}$, it is possible to determine a best choice of corresponding features c^*. A local assignment procedure based upon proximity of features under transformation, $\mathcal{F}$, is used to determine c^*.

While increasing the number of potentially matching features increases the size of the correspondence space exponentially, the dimensionality of the coregistration space is fixed: $\Re^8$ for the case treated here. Consequently, search in coregistration space has an advantage over search in correspondence space. The term **coregistration-space search** is used to refer to this approach.

3.1 Coregistration-Space Search

The goal of any type of local search is to minimize some error function through iterative improvement. In coregistration-space search, the error function measures the relationship between the predicted model features and the data. This measurement takes into account both range and optical features, but treats the two cases somewhat differently. For the optical features, the error is a function of the gradient response to a tuned filter for each line segment [SWF95]. For range, the error measure is a function of the Euclidean distance from points on the predicted model sampled surface to their nearest neighbor in the range image data.

The local search itself samples each of the 8 dimensions of the coregistration space about the current estimate. Clearly, the step-size used in this sampling is important. The general strategy implemented moves from coarse to fine sampling as the algorithm converges upon a locally optimal solution. The initial scaling of the sampling interval is determined automatically based upon moment analysis applied to the current model and sensor data sets.

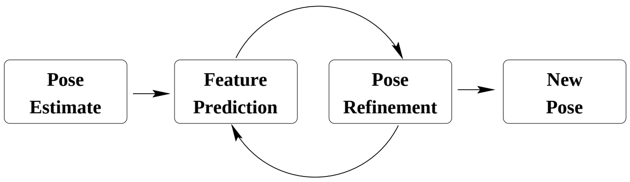


Figure 1: Pose Refinement

The search process forms an iterative generate-and-test loop (Figure 1) in which the current coregistration hypothesis $\mathcal{F}$ is used to predict a set of model features which are in turn used in the error evaluation function. A neighborhood of moves is then examined and the best move, the one with the lowest error, is taken. The features are re-generated for the new coregistration estimate and the process continues.

The neighborhood decouples the 8 dimensions of the coregistration space into three distinct sets of possible moves. The first set represents the 6 dimensions which encode the 3D pose of the sensors relative to the object. The other two represent changes in registration between the two sensors. Search examines pose moves first until a local optimum is reached. Then and only then are changes in sensor registration considered. If a change is made in sensor registration, then additional changes in the pose are again considered. Hence, control alternates between refining pose and refining registration.

When no further progress is possible along any dimension, the resulting 8 values are returned as the locally optimal coregistration estimate. Initial results of the search have shown that the local optima in color space, and the local optima in range space, do not usually coincide. By searching for the model in both the optical and range imagery, local optima in each will be rejected in favor of the global solution.

3.2 Error Terms

The error function to minimize, $E_{\mathcal{M}}(\mathcal{F})$, may be thought of as consisting of two main components: a weighted term representing how well the 3D model line segments fit the current color image, and a weighted term representing how well the sampled surface information fits the range data. These two terms are combined to form the overall match error:

$$E_{\mathcal{M}}(\mathcal{F}) = \alpha_{\mathcal{M}} E_{\mathcal{M},o}(\mathcal{F}) + (1-\alpha_{\mathcal{M}}) E_{\mathcal{M},r}(\mathcal{F}) \tag{2}$$

Each sensor term can be further broken down into two weighted terms: an omission error and a fitness error.

$$E_{\mathcal{M},\mathcal{S}}(\mathcal{F}) = \beta_{\mathcal{S}} E_{fit,\mathcal{S}}(\mathcal{F}) + (1-\beta_{\mathcal{S}}) E_{om,\mathcal{S}}(\mathcal{F}) \tag{3}$$

The subscript $(\mathcal{S})$ is replaced below with o for optical and r for range. The fitness error $E_{fit,\mathcal{S}}(\mathcal{F})$ represents how well the strongest features (as determined by a threshold) match, and the omission error $E_{om,\mathcal{S}}(\mathcal{F})$ penalize the match wherein model features are left unmatched. This happens when no adequate matching features can be found in the sensor data.

3.2.1 Optical Fitness Error: $E_{fit,o}(\mathcal{F})$

The optical fitness error represents how well each model line fits the underlying image. The process of determining the error begins by projecting the predicted 3D model edges into the color image. Projection is possible because both the intrinsic sensor parameters and the approximate location of the target are known. The parameters for the color sensor have been determined off-line using calibration targets [BHP94]. After each edge is projected, a gradient mask tuned to the precise expected orientation is applied to the pixels lying under each line. The gradient response, $\hat{G}_{Line}(k)$, is normalized to the range $[0,1]$. The derivation of the response is presented in [Mar96].

A threshold (v) is used to discard lines with weak responses, and the gradient response is converted to an error term for each line:

$$E_{Line}(k) = \begin{cases} (1-\hat{G}_{Line}(k)) & \hat{G}_{Line}(k) < v \\ 0 & otherwise \end{cases} \tag{4}$$

The fitness error is then formed by summing the error terms, and normalizing by the number of lines l not discarded (lines whose response was set to 0):

$$E_{fit,o}(\mathcal{F}) = \frac{\sum_{k \in ModelLines} E_{Line}(k)}{l} \tag{5}$$

3.2.2 Range Fitness Error: $E_{fit,r}(\mathcal{F})$

The range fitness error represents how well the predicted 3D sampled surface model points fit the actual range data. To reduce computation, only a subset of the range data is examined at a time. When the model sampled surface information is being obtained, the range image plane bounding box is determined. Only those data points lying inside the model bounding box (plus some margin of error) are examined: let this set of data points be (ψ), and let (χ) be the set of all predicted model points.

Computing fitness begins by measuring the Euclidean distance between a single model point $i \in \chi$ and each data point $j \in \psi$. The distance is measured with the model points (χ) placed relative to the data using the current coregistration estimate $\mathcal{F}$:

$$\overline{D}(i,j)_{i \in \chi, j \in \psi} = |i - j| \quad (6)$$

The nearest neighbor of each model point i is that which minimizes the Euclidean distance $\overline{D}$. This distance to the nearest neighbor may be written as:

$$\hat{H}_{point}(i) = \overline{D}(i,j) \quad : \forall k \in \psi \; \overline{D}(i,j) \leq \overline{D}(i,k) \quad (7)$$

The fitness error is then a function of these nearest neighbor distances.

$$E_{point}(i) = \begin{cases} \hat{H}_{point}(i) & H_{point}(i) < \tau \\ 0 & otherwise \end{cases} \quad (8)$$

The threshold (τ) places an upper bound on the distance between matching features, and is set to discard points considered too far away to match. The total fitness for range is then summed over the matched points and normalized to lie in the range $[0,1]$. Normalization takes account of the number of matched points p and the maximum allowable distance τ:

$$E_{fit,r}(\mathcal{F}) = \frac{\sum_{i \in \chi} E_{point}(i)}{p \cdot \tau} \quad (9)$$

3.2.3 Omission Error: $E_{om,\mathcal{S}}(\mathcal{F})$

Omission accounts for weak responses in optical and unmatched points in range. Omission is needed to prevent fixation upon very small numbers of strongly matched features. Omission introduces a bias in favor of accounting for as many model features as possible. The general form of the omission error is:

$$E_{om,\mathcal{S}}(\mathcal{F}) = \begin{cases} \cdot \frac{e^{\alpha \cdot w} - 1}{e^{\alpha} - 1} & \alpha \neq 0 \\ w & \alpha = 0 \end{cases} \quad (10)$$

where w is ratio of unmatched model features over the total number of model features. The parameter α introduces a non-linear bias which essentially reduces the penalty of small amounts of omission while increasing the penalty for large amounts of omission. A detailed explanation of this relationship may be found in [Bev93]. For the optical omission error, $E_{om,o}(\mathcal{F})$, w is the number of unmatched lines over the total number of lines.

For the range data, omission is measured in both directions: model-to-data and data-to-model. Because C allows many model features to match a single range feature, the matching algorithm can be drawn away from the true solution during the first few iterations of the search. Including a term to measure how much range data is omitted from the match corrects this problem. Thus, range omission is given by:

$$E_{om,r}(\mathcal{F}) = \begin{cases} \frac{1}{2} \cdot \left(\frac{e^{\alpha p} - 1}{e^{\alpha} - 1} + \frac{e^{\alpha q} - 1}{e^{\alpha} - 1}\right) & \alpha \neq 0 \\ \frac{p+q}{2} & \alpha = 0 \end{cases} \quad (11)$$

where p is the ratio of unmatched model points over the total number of model points, and q is the number of unmatched data points over the total number of data points.

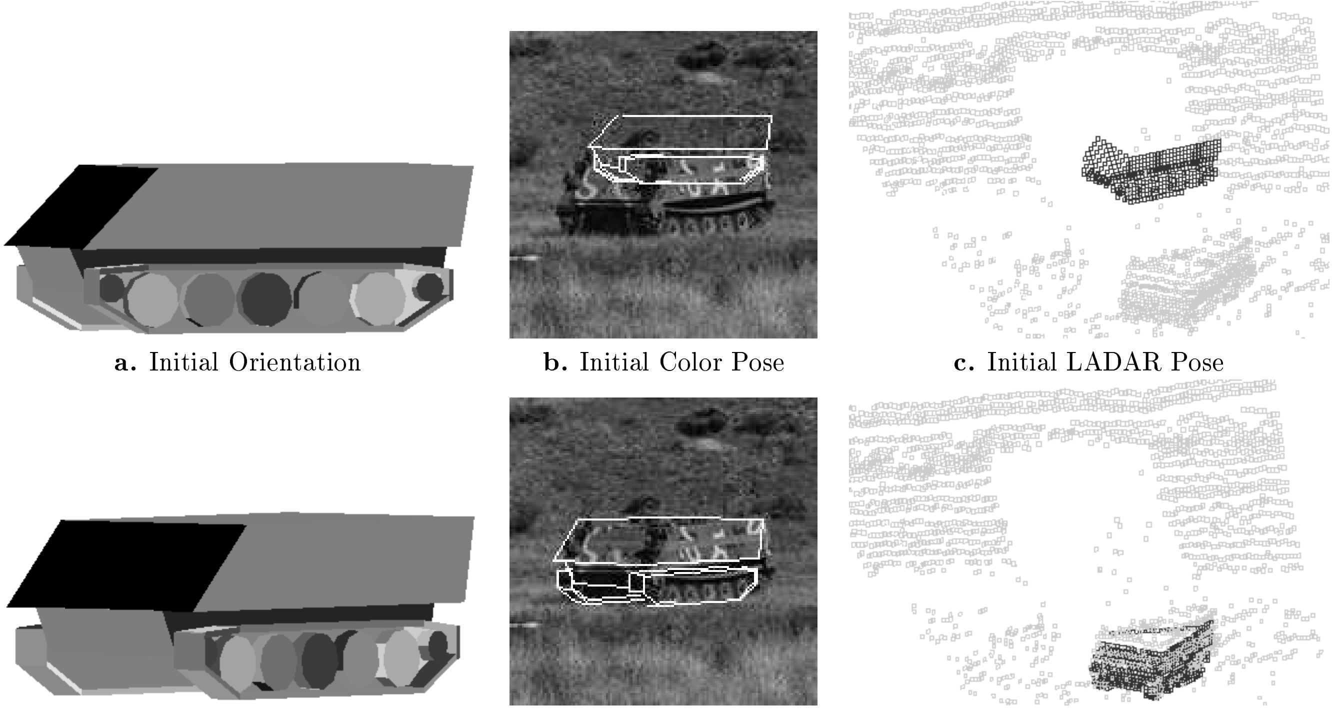
4 Results

Initial testing of the combined feature prediction and matching algorithms is being done on pairs of range and optical imagery from the Fort Carson dataset [BPY94]. Results for two shots with varying level of difficulty are presented. A shot is defined as a pair of approximately registered range and optical images.

The first shot, Shot 20 from Vehicle Array 5, contains an M113 APC sitting in an open field approximately 50 meters from the sensor. This shot is relatively simple and was selected as a proof-of-algorithm example. The other shot, Shot 35 from Vehicle Array 9, shows the same M113 APC side-on at approximately $100m$ with its nose point slightly down relative to the rear of the vehicle.

For each shot, matching is initialized using a coregistration estimate provided by a range template matching algorithm [Bev92]. The estimate provides both an orientation and a translation estimate for the vehicle relative to the two sensors. The template matching algorithm ranks the set of alternative estimates using a confidence factor. These initial hypotheses are needed in order to provide a starting point for our matching algorithm. However, early experiments suggest these hypotheses can be off by as much as 30° in orientation and $25m$ in translation. On both shots, the matching algorithm dramatically improves the initial estimate.

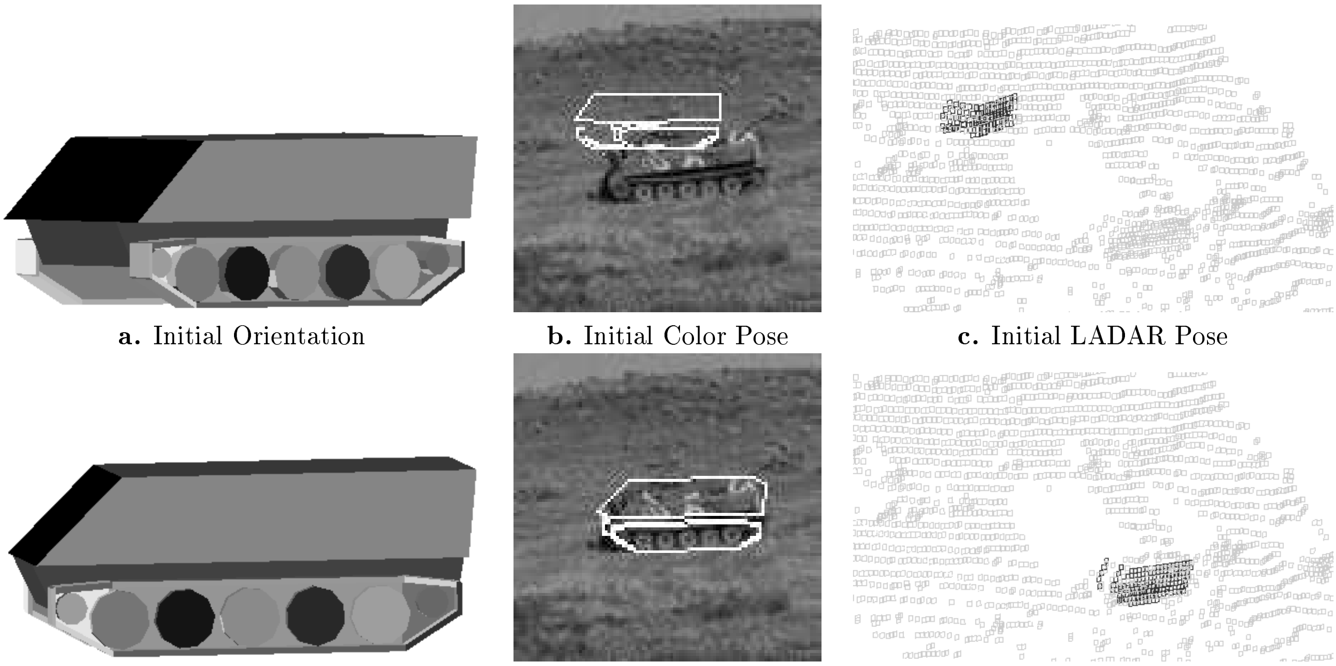
Figures 2 and 3 show both the initial starting conditions and final matches obtained using our combined feature prediction and matching algorithms. Before discussing the matching results themselves, it is helpful to provide some background on the imagery. The color images (shown in black and white here) are 150x150 pixel squares cropped from the complete 720x480 image. For Shot 20, the vehicle was roughly centered at the optical image axis. For Shot 35, the vehicle was roughly 50 pixels off optical axis. The color imagery was obtained with a standard $35mm$ camera. The images were digitized to a Kodak Photo Compact Disks. Predicted



a. Initial Orientation **b.** Initial Color Pose **c.** Initial LADAR Pose

d. Resulting Orientation **e.** Resulting Color Pose **f.** Resulting LADAR Pose

Figure 2: Local Search Results for Shot 20 Array 5 (M113 APC)



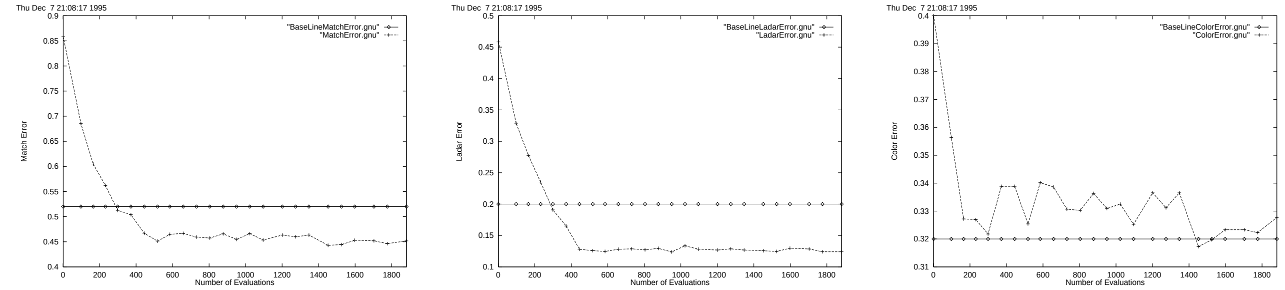
a. Initial Orientation **b.** Initial Color Pose **c.** Initial LADAR Pose

d. Resulting Orientation **e.** Resulting Color Pose **f.** Resulting LADAR Pose

Figure 3: Local Search Results for Shot 35 Array 9 (M113 APC)

model features are shown as white lines, drawn on top of the image.

The range data was obtained with a LADAR sensor. The LADAR ranging device scans a scene in a series of parallel vertical strips, generating a rectangular array of range values with 12 bit resolution. The field of view of the current LADAR system is approximately 15° horizontally and 3° vertically. The maximum range of depth values is approximately $300m$. The images show the range data rendered as a set of 3D hollow polygons



a. $E_{\mathcal{M}}$ **b.** $E_{\mathcal{M},r}(\mathcal{F})$ **c.** $E_{\mathcal{M},o}(\mathcal{F})$

Figure 4: Error Terms for Shot 35 Array 9 (M113 APC)

[GBSF94, GBSF95], from a viewpoint slightly above and to the left of the sensor position. The actual LADAR data is shown in light grey, and the predicted sampled surface features are shown in black[1].

Figures 2a, 2b, 2c show the initial pose estimate in each sensor coordinate system for shot 20. Figures 2d, 2e, 2f show the results of the local search algorithm. As can be seen, the algorithm corrects for a substantial amount of error in the initial estimate. It took the search algorithm approximately 30 moves to come to the solution shown. On the order of 1500 evaluations of the error function were performed. Figure 3 shows the results of the algorithm applied to Shot 35. Again, the algorithm was able to account for a substantial amount of error in the initial estimate.

Due to the interleaving of feature prediction and matching, the match error function does not monotonically decrease. The total error, along with the constituent optical and range components, can be plotted. Figure 4 shows the error plots for Shot 35, with the error term on the vertical axis, and the number of error evaluations on the horizontal axis. Several observations can be made about the graphs. The first is the rapid initial reduction in the error. The second is the relatively jumpy character of the optical error as the algorithm converges to the best set of coregistration parameters. Both trends are side effects of the search strategy alternating between adjustments to the sensor parameters and adjustments to the image registration. These plots also suggest that feature generation and matching are interacting in subtle ways and more study is required to better understand and characterize these interactions. It should be that noted the optical feature set changes significantly from start to finish, which suggests the cause of the choppy error is due to abrupt changes in selected features.

[1] This imagery was collected with a low resolution LADAR and wide angle lens in order to approximate resolutions comparable to targets viewed at 1 to $2km$ with more modern sensors.

5 Conclusion

By combining dynamic model feature prediction and local search in coregistration space, we have demonstrated the ability to find geometrically precise matches between CAD models and multi-sensor data. Moreover, we have done this with real data from a highly difficult object recognition domain.

The time required to perform dynamic feature prediction is relatively small, on the order of one second. When compared to the time required by the iterative search algorithm to test new matches it is not the most significant factor. The benefit of performing scene specific lighting calculations to predict internal structure far outweighs the slight run-time penalties. These internal features allow us to solve problems which have proved unsolvable using silhouette features alone. Future work will refine the radiometric modeling used by this phase of the algorithm: the current version while effective is clearly very simple.

Our algorithm is able to substantially reduce errors and generate visibly improved matches, however some difficulties have been observed getting the algorithm to arrive at precisely the best final solution. For example, while the results shown above in Figure 3 are dramatic improvements over the initial estimate, the nose of the vehicle is still slightly up relative to the imagery. We are examining techniques to improve the search method so as to prevent premature termination, thereby improving the match.

References

[BDHR94] Shashi Buluswar, Bruce A. Draper, Allen Hanson, and Edward Riseman. Non-parametric Classification of Pixels Under Varying Outdoor Illumination. In *Proceedings: Image Understanding Workshop*, pages 1619–1626, Los Altos, CA, November 1994. ARPA, Morgan Kaufmann.

[Bev92] James E. Bevington. Laser Radar ATR Algorithms: Phase III Final Report. Technical report, Alliant Techsystems, Inc., May 1992.

[Bev93] J. Ross Beveridge. *Local Search Algorithms for Geometric Obejct Recognition: Optimal Correspondence and Pose.* PhD thesis, University of Massachuesetts at Amherst, May 1993.

[BHP94] J. Ross Beveridge, Allen Hanson, and Durga Panda. Integrated color ccd, flir & ladar based object modeling and recognition. Technical report, Colorado State University and Alliant Techsystems and University of Massachusetts, April 1994.

[BHP95] J. Ross Beveridge, Allen Hanson, and Durga Panda. Model-based fusion of flir, color and ladar. In Paul S. Schenker and Gerard T. McKee, editors, *Proceedings: Sensor Fusion and Networked Robotics VIII*, Proc. SPIE 2589, pages 2 – 11, October 1995.

[BPY94] J. Ross Beveridge, Durga P. Panda, and Theodore Yachik. November 1993 Fort Carson RSTA Data Collection Final Report. Technical Report CSS-94-118, Colorado State University, Fort Collins, CO, January 1994.

[BSS96] J. Ross Beveridge, Mark R. Stevens, and N. A. Schwickerath. Toward target verification through 3-d model-based sensor fusion. *IEEE Transactions on Image Processing*, page (Submitted), 1996.

[CA87] C. H. Chien and J. K. Aggarwal. Shape recognition from single silhouettes. In *International Conference on Computer Vision*, pages 481–490, 1987.

[CS94] Jin-Long Chen and George C. Stockman. Determining pose of 3d objects with curved surfaces. Technical Report CPS-93-40, Michigan State University, 1994.

[CSR93] Jin-Long Chen, George C. Stockman, and Kashi Rao. Recovering and tracking pose of curved 3d objects from 2d images. In *Proceedings Computer Vision and Pattern Recognition*, pages 233–239, June 1993.

[GBSF94] Michael E. Goss, J. Ross Beveridge, Mark R. Stevens, and Aaron Fuegi. Visualization and Verification of Automatic Target Recognition Results Using Combined Range and Optical Imagery. In *Proceedings: Image Understanding Workshop*, pages 491–494. ARPA, nov 1994.

[GBSF95] Michael E. Goss, J. Ross Beveridge, Mark R. Stevens, and Aaron D. Fuegi. Three-dimensional visualization environment for multi-sensor data analysis, interpretation, and model-based object recognition. In Georges G. Grinstein and Robert F. Erbacher, editors, *Proceedings: Visual Data Exploration and Analysis II*, pages 283–291. SPIE Vol. 2410, feb 1995.

[HH94] Anthony Hoogs and Douglas Hackett. Model-supported exploitation as a framework for image understanding. In *Proceedings: Image Understanding Workshop*, pages 265–268. ARPA, nov 1994.

[Ike87] Katsushi Ikeuchi. Precompiling a geometrical model into an interpretation tree for object recognition in bin-picking tasks. In *Proc. DARPA Image Understanding Workshop*, pages 321–330, February 1987.

[J. 96] J. Ross Beveridge and Bruce A. Draper and Kris Siejko. Progress on Target and Terrain Recognition Research at Colorado State University. In *Proceedings: Image Understanding Workshop*, page (to appear), Los Altos, CA, February 1996. ARPA, Morgan Kaufman.

[KD87] Matthew R. Korn and Charles R. Dyer. 3D Multiview Object Representations for Model-Based Object Recognition. *Pattern Recognition*, 20(1):91–103, 1987.

[Koe84] J.J. Koenderink. What does occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.

[KvD76] J. J. Koenderink and A. J. van Doorn. The Singularities of Visual Mapping. *Biological Cybernetics*, 24:51–59, 1976.

[KvD79] J. J. Koenderink and A. J. van Doorn. The Internal Representation of Shape with Respect to Vision. *Biological Cybernetics*, 32:211–216, 1979.

[LT90] Cheng-Hsiung Liu and We-Hsiang Tsai. 3d curved object recognition from multiple 2d camera views. *Computer Vision, Graphics and Image Processing*, 50:177–187, 1990.

[Mar77] David Marr. Analysis of occluding contour. *Proceedings of the Royal Society of London*, B197:441–475, 1977.

[Mar96] Mark R. Stevens and J. Ross Beveridge. Optical Linear Feature Detection Based on Model Pose. In *Proceedings: Image Understanding Workshop*, page (to appear), Los Altos, CA, February 1996. ARPA, Morgan Kaufman.

[PD87] Harry Platinga and Charles Dyer. Visibility, Occlusion, and the Aspect Graph. Technical Report 736, University of Wisconsin - Madison, December 1987.

[PHK91] J. Ponce, A. Hoogs, and D.J. Kriegman. On using cad models to compute the pose of curved 3d objects. *DACBV*, 91:136–145, 1991.

[Pla88] William Harry Plantinga. *The ASP: A Continuous, Viewer-Centered Object Representation for Computer Vision*. PhD thesis, University of Wisconsin at Madison, 1988.

[PP76] G.W. Paltridge and C.M.R Platt. *Radiative Processes in Meteorology and Climatology*. Elsevier Scientific Publishing Company, 1976.

[SB94] Anthony N. A. Schwickerath and J. Ross Beveridge. Model to Multisensor Coregistration with Eight Degrees of Freedom. In *Proceedings: Image Understanding Workshop*, pages 481 – 490, Los Altos, CA, November 1994. ARPA, Morgan Kaufmann.

[SBG95] Mark R. Stevens, J. Ross Beveridge, and Michael E. Goss. Reduction of BRL/CAD Models and Their Use in Automatic Target Recognition Algorithms. In *BRL/CAD Symposium 95*, 1995.

[SD92] W. Brent Seales and Charles R. Dyer. Modeling the Rim Appearance. In *Proceedings of the 3rd International Conference on Computer Vision*, pages 698–701, 1992.

[Ste95] Mark R. Stevens. Obtaining 3D Silhouettes and Sampled Surfaces from Solid Models for use in Computer Vision. Master's thesis, Colorado State University, September 1995.

[SWF95] G.D Sullivan, A.D. Worrall, and J.M Ferryman. Visual Object Recognition Using Deformable Models of Vehicles. In *Workshop on Context-Based Vision*, pages 75–86, june 1995.

[U. 91] U. S. Army Ballistic Research Laboratory. *BRL-CAD User's Manual*, release 4.0 edition, December 1991.

[WI93] M.D. Wheeler and K. Ikeuchi. Sensor modeling, markov random fields, and robust localization for recognizing partially occluded objects. *IUW*, 93:811–818, 1993.

[WMA84] Y. F. Wang, M. J. Magee, and J. K. Aggarwal. Matching three-dimensional objects using silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:513–518, 1984.

[WW80] T.P. Wallace and P.A. Wintz. An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. *Computer Graphics and Image Processing*, 13:99–126, 1980.